# zzstructure systems

## Trademarked in USA as ZigZag®
## (with MorphableND™ graphics)

## U.S. Patent no. 6,262,736.

## A new kind of software for database, visualization, programming and all other purposes.

### A New Paradigm for Users-- NO FILES, HIERARCHIES OR APPLICATIONS

We hear about "new software paradigms" all the time, but most of them are pretty local. Zzstructure is sweepingly different. To those in the zzstructure world, MS-DOS, Windows, Linux, Mac and Palm are all pretty much the same.

ZZstructure projects are presently underway in the USA, United Kingdom, Finland, Japan and Australia. (Only those under Ted's personal direction get the name ZigZag.)

ZZstructure begins from first principles. It builds up from very simple rules (a "cosmology") which has been carefully architected to generate an entire computer universe, with many hidden adaptations and adjustments. It is intended for the entire range of computer function, across desktop machines, PDAs, cellphones and on-line services.

The zzstructure design includes equivalents to--
- database, spreadsheet
- word processing-- but with complex document organization and editing made easy
- unification / synchronization of all a person's data on all machines
- on-line applications, use via Web browsers
- use in cellphones

### A New Paradigm of Mechanism, Simpler Underneath

As an internal engine for software, ZigZag is very different from existing software paradigms (esp. object-oriented programming).

### New Powers for Users

- New visualizations (which can be easily added to)
- Easy database
- "Spreadsheet" no longer separate from the rest of software; dynamic cells easily embeddable in text or elsewhere (with considerably more ease than OLE)

- Easy programming, since connection is already built in (75% of programming is about string matching to line up connections)

## Wild New Graphics (Not yet implemented)

User graphics will be based on multidimensional morphing continuous coordinates (MorphableND, pronounced "morphablend") allowing every ZigZag object and space-point to have as many coordinates, on as many spatial dimensions, as you like. This means that both objects and the viewing space itself can morph dynamically into new arrangements and new perspectives. But this will project onto today's 3D graphics boards, giving them new powers that have been previously unsuspected.

---

## Several Simple Explanations of zzstructure, all true, all different

0) Crossed lists in as many dimensions as you like. (You may rapidly hop an item up and down any of the lists it's on.)

1) All data consists of cells in rows and columns.

  - This simplifies viewing and navigation
  - This simplifies programming
  - Views are interchangeable (since all data has the same format)
  - Menus are built in (simply as rows and columns of executable cells)
  - Everything can be navigated fast with a cellphone keyboard, or in two windows with Nintendo-type controls
  - Interfaces for the vision-impaired and hearing-impaired maintain the same structure.

2) ZZstructure represents all relations as discrete (logical) dimensions. Every relation (or class of relations) is a dimension, and the view can be rotated to show different sets of dimensions. This accomplishes an amazing amount.

3) Free-floating construction blocks! Think of it as Tinkertoys(R), where balls are connected by sticks at certain angles. Everything is built with cells and untyped connectors at particular angles; each angle we will now call a dimension.

4) Spreadsheet cells sprung free! A spreadsheet is rectangular, because you're going to print it out. A spreadsheet has empty cells because you're going to print it out. Relax these constraints, but keep--

- THE SPREADSHEET CELL, which has one context vertically and another context horizontally.
- But let's generalize it. A cell can have at most two neighbors in any dimension, FOR AS MANY DIMENSIONS AS YOU LIKE.
- And you may create a new dimension instantly, on the fly.

Now instead of a rectangular spreadsheet we get • crystals, • wheels, • staircases, • secret passages, and gosh knows what we haven't seen yet.

5) A wind-up crossword puzzle! Programmatic operations actually set the structure in motion.

6) An aesthetic of simplicity, elegance, consistency and space. I also find its paradoxes-- that is, violations of our spatial expectations-- somewhat humorous.

### No Division into 'Applications'

Instead of separate applications, we have "applitudes" (zones of related content) for which we create special dimensions and other mappings. But these may be shared. The relations built for one applitude (example: genealogy handled by two dimensions, d.marriage and d.children) can be used anywhere else (d.marriage and d.children may also connect names in your address-book applitude).

### EASY EVERYDAY DATABASE, ALL INTERCONNECTED

A user has no separate databases, but one interconnected plex.

Example: An entry for a man in the address book can connect to--

- his address, phone number, email address
- his website
- how much he owes you (or you owe him)
- his wife, children, parents-- and the same information for them
- work you're expecting from him next week
- what you have to remember to tell him
- any other relationship you want to represent.

Example: All records of possessions, their warranties and repair history, receipts, what boxes they're in, may be in a single unified structure.

ZigZag is much more forgiving than conventional databases. Parallel (corresponding) records may have different combinations of fields. Cells, entries, new fields (columns) are easily created.

## The Overall ZigZag Design

The two prototypes so far have implemented only about 1/10 of the functionality in the general design.

The general zzstructure design (as written up in 1999) may be found at http://xanadu.com/zigzag/fw99/. The present design has evolved considerably since that writeup, and is presently being documented and spec'd.

To Ted Nelson Home Page

02.05.23

zzStruxSys

( d6z

•